Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 29 (2012) 301 – 305

Procedia Engineering

www.elsevier.com/locate/procedia

2012 International Workshop on Information and Electronics Engineering (IWIEE)

High Performance VLSI Architecture of Bi-directional Motion Search for B Picture in H.264

Mei-Hua Gu[a], Rui Kong[b,*]

[a]*College of Electronic Information, Xi'an Polytechnic University, Xi'an, 710048, P.R. China*
[b]*Department of Electronics Engineering, Xi'an University of Technology, Xi'an, 710048, P.R. China*

Abstract

Bi-directional motion search is one of the important features of B picture coding in H.264/AVC. However, its high computational complexity and huge memory traffic make design difficult. This paper proposes a high throughput and cost efficient VLSI architecture for bi-directional integer motion estimation (Bi-IME). The redundancy of the joint motion search is removed, and the algorithm is simplified. Novel memory structure and intelligent reading method are designed to satisfy the iterations of full search with two reference windows. The parallel and sequence techniques are adopted to process the matching procedure. After logic synthesis using SMIC 0.13 μm standard cell library, under a clock frequency of 300MHz, the proposed Bi-IME architecture can provide processing capacity up to 149M MBs/sec which is enough for 1080p real-time video systems.

Keywords: H.264/AVC ; bi-directional ; joint motion estimation ; VLSI

1. Introduction

H.264/AVC [1,2]provides three traditional prediction methods, the first one is forward prediction, the second one is backward prediction, the third one is basic bi-directional prediction. It is noteworthy that the compensated signals used in the above three predictions are all obtained from uni-motion estimation, including independent forward ME and backward ME. In addition, B picture coding in H.264/AVC also support the other two prediction methods, whose prediction signals are the linear combination of the

* * Corresponding author. Tel.:+86-015309255317;
E-mail address:gu_meihua@163.com.

1877-7058 © 2011 Published by Elsevier Ltd. Open access under CC BY-NC-ND license.
doi:10.1016/j.proeng.2011.12.711

compensated signals from the adjacent past and future reference frames[3]. The new bi-directional prediction is a joint estimation for the forward and backward motion vectors[4], aims at obtaining a pair of MVs by bi-directional motion search in the past and future reference frames. The two main steps included in the new bi-directional motion estimation are bi-directional integer motion estimation (Bi-IME) and bi-directional fractional motion estimation (Bi-FME). Compared with uni-directional ME, bi-directional motion search is more complicated, and presents a formidable computational challenge. Bi-directional ME forms a major computational bottleneck in video processing applications.

Because of the intensive computation of Bi-directional ME, the hardware accelerator is critical for the real time encoding system, especially for HDTV applications. However, the ME VLSI architectures found from the exist literature are all for uni-directional ME. Therefore, the VLSI architecture for bi-directional ME proposed in this paper is the first presentation of this field. In the following sections, the algorithm and VLSI architecture of Bi-IME will be described in detail.

2. Bi-IME Algorithm

2.1. The search region

There are four iterations in Bi-IME, a pair of reference image block is needed for each Bi-IME procedure, one is from the main search region, and the other is from the auxiliary reference block. Bimv is exploited to determine the search region in the reference from list^1, mv is exploited to determine the search region in the reference from list. The related search areas are shown in Fig.1. In the first Bi-IME iteration, the main search region is from the reference in list^1, and is a 33×33 search window centered by the position bimv points to, the auxiliary reference block is from the position mv points to in the reference of list. In the second Bi-IME iteration, the main search region is from the reference in list, and is a 17×17 search window centered by the position mv points to, while the auxiliary reference block is from the position bimv points to in the reference of list^1. In the third Bi-IME iteration, the main search region is from the reference in list^1, and is a 9×9 search window centered by the position bimv points to, and the auxiliary reference block is from the position mv points to in the reference of list. In the last Bi-IME iteration, the main search region is from the reference in list, and is a 5×5 search window centered by the position mv points to, and the auxiliary reference block is from the position bimv points to in the reference of list^1.

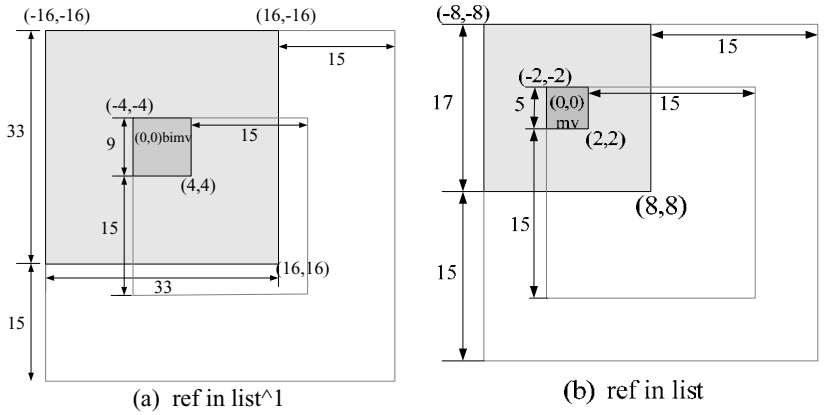

Fig. 1. Search regions for Bi-ME, (a) Reference in list^1, (b) Reference in list

2.2. Full search Bi-IME

The reference image of Bi-IME is from the reference frames in both the forward and the backward directions, which is called the main/auxiliary reference frame. The two kinds of references are exchanged several times, and is controlled by the parameter of iter. During the Bi-IME procedure, all the possible candidate blocks in the main reference frame are considered, and are taken as the first reference signal during the matching procedure. While the second reference image block is always the same from the fixed position in the auxiliary reference. The prediction signal is the linear average of the signals from the main reference block and the auxiliary reference block.

The two reference region are obtained from list0 and list1, the predict pixel is the average of the two pixels from list0 and list1, as (1).

$$pred(i,j) = (pred0(i,j) + pred1(i,j) + 1) >> 1 \quad (1)$$

Where pred0(i, j) and pred1(i, j) are the predict pixels from the references of list0 and list1, pred(i, j) is the constructed pixel of Bi-IME.

The matching criterion used in Bi-IME is as following:

$$SAD(dx,dy) = \sum_{m=x}^{x+M-1} \sum_{n=y}^{y+N-1} \left| I_{curr}(m,n) - I_{ref}(m+dx, n+dy) \right| \quad (2)$$

$$(bimv_x, bimv_y) = (dx,dy)\big|_{\min(SAD(dx,dy)+mvd\_bits(dx,dy))} \quad (3)$$

Where $I_{curr}(m,n)$ is a pixel at the position (m, n) in the current MxN block, $I_{ref}(m+dx,n+dy)$ is the corresponding reference constructed pixel. (*dx, dy*) is a candidate mv from the main reference searching area. mvd_bits(dx,dy) is the sum of bit rates of generated from the two mv differences.

3. Bi-IME VLSI architecture

3.1. Memory

The memory utilization in Bi-IME is described first. In mode of 16×16, the two reference windows size are 32×32 and 24×24, the memory design are as Fig.2. Fig.2 shows the adopted storage combination, one piece saves the high half of one line data in the search window; the other piece saves the low half data. Each line of data from SW0 and SW1 must be read every time.

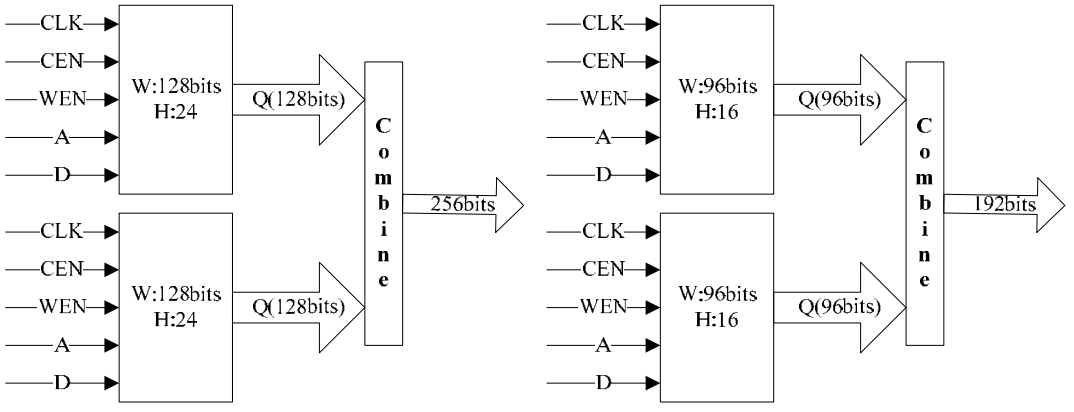

Fig.2. Memory design of SW0 and SW1

3.2. Read control

When the data in SW0 and SW1 are ensured, Bi-IME computation can be performed. The main reference and the auxiliary reference will exchange at most 3 times. When iter equals to 1, there are 17 matching positions in the vertical directions of SW0, 32 pixels at one line are read each cycle, 16 lines can be finished within 16 cycles, as shown in Fig.3. For 16×16 mode, each vertical matching line corresponds to 16-pixel high, and can be finished within 16 cycles, vert line0~vert line 16 denote 17 vertical position line. The pixel block at the fixed position in SW1 are read when iter equals to 1, SW1 is the auxiliary reference image.

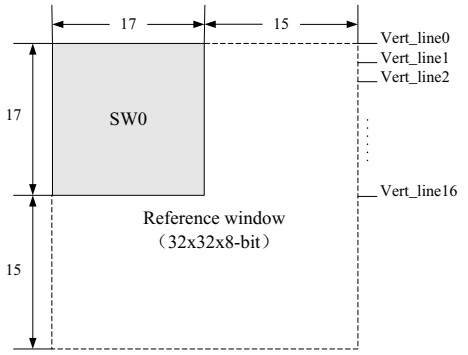

Fig.3. Reading method of matching data when iter equals to 1

3.3. Computing and matching module

The computation unit is responsible for the matching, and computing the cost of SAD at each position, then compare them to get the best matching position, the structure is shown as Fig.4.

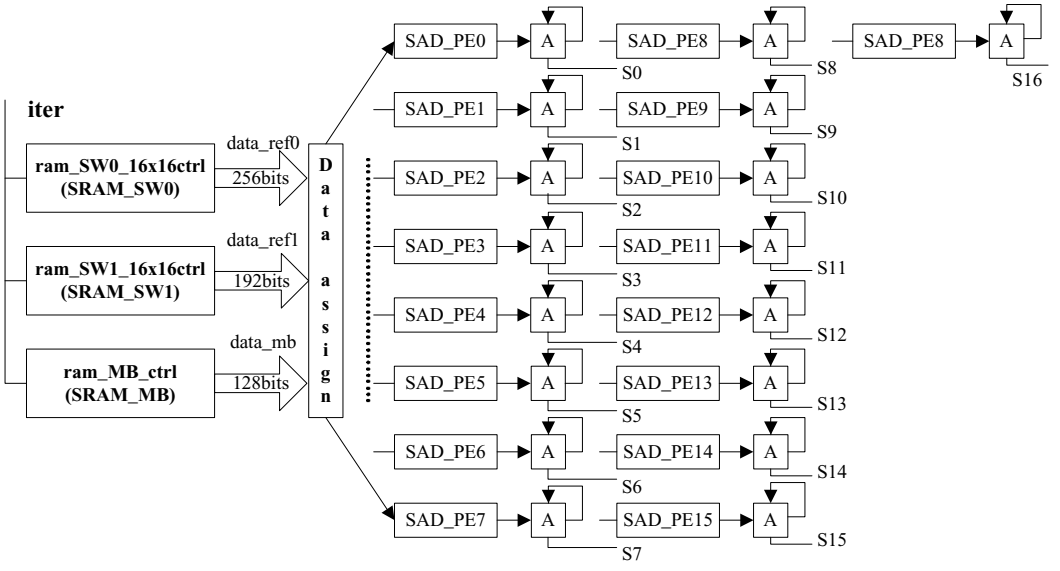

Fig.4. Structure of the matching curcuit

When the read signal enables, three kinds of data are provided from SRAM_SW0, SRAM_SW1, and SRAM_MB, the data width of data_ref0 and data_ref1 is [255:0] and [191:0], and store one line of data from SW0 and SW1 respectively. The data width of data_mb is [127:0].

4. Experimental results

The proposed Bi-IME architecture is described and verified with Verilog-HDL in VSC environment and synthesized by Synopsis Design Compiler using SMIC 0.13μm CMOS standard cell library. The architecture of Bi-IME uses three search windows with the sizes of 17×17, 9×9 and 5×5. Bi-directional motion search for mode 16x16, 16x8, and 8x16 is processed sequentially, while 2 16x8 blocks for mode 16×8, 2 8×16 block for mode 8x16 are processed in parallel. After synthesis, the maximum operating clock frequency is 300 MHz with about 112k gates except SRAM. The total size of SRAM is 9K bytes. Under a clock frequency of 300MHz, the architecture allows the real-time processing of 1920 ×1080(1080p) at 18k fps. Table 1 shows the performance of the proposed Bi-IME architecture.

Table 1. Performance of Bi-IME architecture

Algorithm	Bi-directional full search integer motion estimation
Search range	17×17, 9×9, 5×5
Block size	16×16, 16×8, 8×16
Technology	SMIC 0.13μm CMOS
Gate counts	112k
SRAM on chip	24×128×2-bit, 36×96 ×2-bit
Max frequency	300MHz
Throughput	149M MBs/sec

5. Conclusions

This paper presents an efficient Bi-IME architecture. It can support the highest specification with the largest search range, and the design presented exhibits good performance in terms of throughput, gate counts and other aspects catered for. From the dynamic simulation, the whole design of Bi-IME can provide processing capacity up to 149M MBs/sec which is enough for 1080p real-time video streams.

References

[1] JVT. Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification,2005.

[2] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and et al. Video coding with H.264/AVC:tools, performance, and complexity, *IEEE Circuits and Systems Magazine*,2004, 4:7-28.

[3] Markus Flierl, Bernd GirodGeneralized. B Pictures and the draft H.264/AVC video-compression standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 2003, 13(7):587-597.

[4] Siu Wai Wu, Allen Gersho. Joint estimation of forward and backward motion vectors for interpolative prediction of video. IEEE Transactions on Image Processing, 1994, 3(5): 684-687.

[5] JVT. Reference Software JM16.1. *http://iphome. hhi.de/ suehring/ tml/ download/ old_jm/ jm16.1.zip*, 2009.